PERGAMON

Journal of Quantitative Spectroscopy & Radiative Transfer 63 (1999) 15–29

Journal of Quantitative Spectroscopy & Radiative Transfer

An efficient nonlinear solution method for non-equilibrium radiation diffusion

D.A. Knoll*, W.J. Rider, G.L. Olson

Applied Theoretical and Computational Physics Division, Los Alamos National Laboratory, XHM M.S. D413, Los Alamos, New Mexico 87545, USA

Received 10 June 1998

Abstract

A new nonlinear solution method is developed and applied to a non-equilibrium radiation diffusion problem. With this new method, Newton-like super-linear convergence is achieved in the nonlinear iteration, without the complexity of forming or inverting the Jacobian from a standard Newton method. The method is a unique combination of an outer Newton-based iteration and and inner conjugate gradient-like (Krylov) iteration. The effects of the Jacobian are probed only through approximate matrix–vector products required in the conjugate gradient-like iteration. The methodology behind the Jacobian-free Newton–Krylov method is given in detail. It is demonstrated that a simple, successive substitution, linearization produces an effective preconditioning matrix for the Krylov method. The efficiencies of different methods are compared and the benefits of converging the nonlinearities within a time step are demonstrated.

Keywords: Radiation diffusion; Non-equilibrium; Newton–Krylov methods

1. Introduction

Radiation transport in astrophysical phenomena and inertial confinement fusion is often modeled using a diffusion approximation. When the radiation field is not in thermodynamic equilibrium with the material then a coupled set of time dependent diffusion equations is used to simulate energy transport [1]. These equations are highly nonlinear and exhibit multiple time and space scales. Implicit integration methods are desired to overcome undesirable time step

*Corresponding author. Tel.: 001-505-667.7467; fax: 001-505-667-3726; e-mail: nol@lanl.gov

0022-4073/99/$-see front matter
PII: S0022-4073(98)00132-0

restrictions. Traditionally, the coupling of these systems has been handled via operator splitting, and the nonlinearities are seldom converged within a time step [1]. Both of these choices impose "effective" time step size restrictions for accuracy and nonlinear stability. Additionally, the nonlinear residual of the system is not formed, and thus it cannot be used to monitor convergence within a time step. We are motivated to begin to quantify the effects of converging, or not converging, the nonlinearities within a time step. Furthermore, we are motivated to develop an efficient iterative method for converging the nonlinearities within a time step.

We present the results of applying a Jacobian-free Newton-GMRES method [2] to a non-equilibrium radiation diffusion problem. The Generalized Minimal RESidual (GMRES) [3] algorithm is a conjugate gradient-like (Krylov) linear iterative method. In the Newton–GMRES (Newton–Krylov) method the effects of the Jacobian can be approximated through matrix–vector products, which are required in the Krylov method. This Newton–Kryolv method can thus provide Newton-like nonlinear convergence without requiring the actual formation or inversion of the Jacobian matrix. This methodology has been applied to a variety of boundary value problems with complex, multiple time-scale, physics [4–7]. Here we apply the methodology to a time-dependent problem in non-equilibrium, flux-limited, radiation diffusion. It is demonstrated that, on this problem, a simple linearization, successive substitution [8], produces a effective preconditioning matrix for the Krylov method [9]. The methodology behind the Jacobian-free Newton–Krylov method is given in detail. The efficiency of the proposed Newton–Krylov nonlinear iterative method is compared to a more standard nonlinear iterative method, successive substitution. The benefits of converging the nonlinearities within a time step are examined.

We wish to emphasize that the proposed approach is different from other, more standard, Newton-based methods applied to this type of problem [10, 11]. Our proposed method does not require the formation of the Jacobian, yet obtains Newton-like nonlinear convergence rates. The details of evaluating the Jacobian are not given in Ref. [11]. In Ref. [11] an outer Newton-based iteration is used with an inner GMRES iteration, but apparently not in a matrix-free fashion. In Ref. [10] the details of evaluating the Jacobian are given.

2. Physics model and spatial discretization

Ignoring hydrodynamics and thermal conduction and working in one dimension results in the following coupled system for radiation energy, E, and material temperature, T.

Radiation diffusion (grey approximation):

$$\frac{\partial E}{\partial t} - \frac{\partial}{\partial x}\left(cD_r \frac{\partial E}{\partial x}\right) = c\sigma_a(aT^4 - E). \tag{1}$$

Material energy balance:

$$C_v \frac{\partial T}{\partial t} = -c\sigma_a(aT^4 - E). \tag{2}$$

Here C_v is the material specific heat, c is the speed of light, a is the Stefan–Boltzmann constant, and σ_a is the photon absorption cross-section. In thermal equilibrium we have $E = aT^4$, and for the non-equilibrium case one can define a radiation temperature as, $T_r = (E/a)^{0.25}$. For simplicity, and

to be consistent with the previous model problems of [12, 13] we will work in an arbitrary system of units where $C_v = c = a = 1.0$. Additionally the form $\sigma_a = T^{-3}$ will be used for the absorption cross-section. It has been assumed here that the scattering cross-section, σ_s is zero. From simple isotropic diffusion theory [1] the following form for the radiation diffusion coefficient results:

$$D_r(T) = \frac{1}{3\sigma_a}. \tag{3}$$

However, in regions of strong gradients simple diffusion theory can fail, resulting in a flux of energy moving faster than the speed of light. To prevent this artificial behavior the diffusion coefficient can be augmented in a fashion referred to as flux limiting [1]. The performance of the proposed algorithm will be investigated with two different functional forms for the flux limiting. The first method has the functional form,

$$D_r(T, E) = \frac{1}{(3\sigma_a + 1/E|\partial E/\partial x|)}. \tag{4}$$

This flux-limiting method will be referred to as method one. Discussion and use of this type of flux limiter can be found in Refs. [14, 15]. There are several ways to implement this flux-limiter, and the specific implementation used in this paper will be given shortly. The second method originates from Ref. [16] and has the functional form

$$D_r(T, E) = \frac{\lambda(T, E)}{\sigma_a \omega(T, E)}. \tag{5}$$

This flux-limiting method will be referred to as the Levermore method. For details on this method, specifically definitions of λ and ω, see Ref. [16]. We will provide our discrete implementation form for both methods.

The model problem considered in this study is taken from [12, 13] and consists of a unit radiation flux impinging on an initially cold slab of unit depth. This results in mixed, or Robin, boundary conditions at $L = 0$ and $L = 1$. Following [12], at $L = 0$ we use

$$\frac{1}{4}E - \frac{1}{6\sigma_a}\frac{\partial E}{\partial x} = 1, \tag{6}$$

and at $L = 1$ we use,

$$\frac{1}{4}E + \frac{1}{6\sigma_a}\frac{\partial E}{\partial x} = 0. \tag{7}$$

Second order in space finite volumes is used to discretize the diffusion operator in the radiation diffusion equation on a uniform grid. The diffusion coefficient, D_r, is evaluated at cell faces by linearly interpolating T and E to cell faces and then evaluating D_r. This is, of course, inadequate for multi-material problems. Considering a uniform grid, the spatial discretization of the diffusion term in finite volume i is then,

$$\int_{vol,i} \frac{\partial}{\partial x}\left(D_r \frac{\partial E}{\partial x}\right) \mathrm{d}vol_i = \frac{D_{r,i+1/2}(E_{i+1} - E_i) - D_{r,i-1/2}(E_i - E_{i-1})}{\Delta x}. \tag{8}$$

This results from the cell volume, vol, being equal to Δx in one dimension.

Two forms for the flux limited diffusion coefficient are considered. The first one, method one, is implemented as

$$D_{\mathrm{r},i+1/2} = \left[3((T_i + T_{i+1})0.5)^{-3} + \frac{2|E_{i+1} - E_i|}{\Delta x(E_{i+1} + E_i)} \right]^{-1}. \tag{9}$$

Recall that $\sigma_{\mathrm{a}} = T^{-3}$. The second form, the Levermore method, is implemented as

$$\omega_{i+1/2} = + \frac{(0.5(T_{i+1} + T_i))^4}{0.5(E_{i+1} + E_i)}, \tag{10}$$

$$R_{i+1/2} = + \frac{2|E_{i+1} - E_i|}{\Delta x(T_{i+1} + T_i)}, \tag{11}$$

$$\lambda_{i+1/2} = \frac{2 + R_{i+1/2}}{6 + 3R_{i+1/2} + R_{i+1/2}^2}, \tag{12}$$

$$D_{\mathrm{r},i+1/2} = \lambda_{i+1/2}[\omega_{i+1/2}((T_i + T_{i+1})0.5)^{-3}]^{-1}. \tag{13}$$

It is not our motivation here to compare solution results from these two flux-limiting strategies. We will concentrate on comparing solution algorithm performance only.

3. Time integration and nonlinear iteration

Three implicit time integration methods are compared, with all three being based on a first-order backward Euler time integration. The differences between the methods are a result of converging or not converging the nonlinearities, and the method used to converge the nonlinearities. The first method is a fully coupled, linearized, semi-implicit technique which does not converge nonlinearities. By this we mean that D_{r} and σ_{a} are evaluated at previous time step solutions, and T^4 is linearized, such that we have a linear problem at each time step. This method may also be referred to as linearly implicit. The second method uses a successive substitution iteration, also referred as a Picard iteration [8], to converge the nonlinearities at each time step. The third method uses a matrix-free Newton–Krylov method to converge the nonlinearities at each time step.

In the semi-implicit (SI) method all nonlinearities are evaluated at old time step level. Since there is no nonlinear iteration in this method, we only require a time step index, n. Our system of equations becomes

$$\frac{E^{n+1} - E^n}{\Delta t} - \frac{\partial}{\partial x}\left(D_{\mathrm{r}}^n \frac{\partial E^{n+1}}{\partial x} \right) = \sigma_{\mathrm{a}}^n (T^{n+1}(T^n)^3 - E^{n+1}), \tag{14}$$

$$\frac{T^{n+1} - T^n}{\Delta t} = -\sigma_{\mathrm{a}}^n (T^{n+1}(T^n)^3 - E^{n+1}). \tag{15}$$

Here we have linearized $(T^{n+1})^4 \approx T^{n+1}(T^n)^3$. We have also used the linearization $(T^{n+1})^4 \approx (T^n)^3(4T^{n+1} - 3T^n)$ and found that it made no difference on the results presented here.

In the fully implicit Picard (FIP) method nonlinearities are converged within a time step, and all nonlinearities are evaluated at the previous nonlinear iteration level $k-1$. This results in the following system of equations:

$$\frac{E^{n+1,k}-E^n}{\Delta t}-\frac{\partial}{\partial x}\left(D_{\mathrm{r}}^{n+1,k-1}\frac{\partial E^{n+1,k}}{\partial x}\right)=\sigma_{\mathrm{a}}^{n+1,k-1}(T^{n+1,k}(T^{n+1,k-1})^3-E^{n+1,k}), \tag{16}$$

$$\frac{T^{n+1,k}-T^n}{\Delta t}=-\sigma_{\mathrm{a}}^{n+1,k-1}(T^{n+1,k}(T^{n+1,k-1})^3-E^{n+1,k}). \tag{17}$$

Since the initial guess for E^{n+1} and T^{n+1} is E^n and T^n, one iteration of FIP is equivalent to the SI method if the linear solve is converged to a significant tolerance.

In the fully implicit Newton–Krylov (NK) method we again converge the nonlinearities within a time step thus we need both a time step index, n, and a nonlinear iteration index, k. The first-order accurate time integration method is;

$$\frac{E^{n+1,k}-E^n}{\Delta t}-\frac{\partial}{\partial x}\left(D_{\mathrm{r}}^{n+1,k}\frac{\partial E^{n+1,k}}{\partial x}\right)=\sigma_{\mathrm{a}}^{n+1,k}((T^{n+1,k})^4-E^{n+1,k}), \tag{18}$$

$$\frac{T^{n+1,k}-T^n}{\Delta t}=-\sigma_{\mathrm{a}}^{n+1,k}((T^{n+1,k})^4-E^{n+1,k}). \tag{19}$$

The nonlinear iteration is implemented with an inexact, matrix-free Newton–Krylov method. By inexact Newton we mean that the convergence tolerance of the linear solver (GMRES) is proportional to the current nonlinear residual. This same linear convergence tolerance algorithm is used in the FIP method and will be discussed in more detail in the next section. By matrix-free Newton–Krylov we mean that the required Jacobian-vector product in each GMRES iteration is replaced by a finite difference approximation to the true Jacobian-vector product [2]. This concept will also be presented in detail in the next section.

Since the nonlinear functions play such an important role in describing the algorithm, and monitoring convergence, we define them so as to avoid any confusion. The nonlinear functions are the discretized equations at each grid cell. The functions for energy at cell i, F_i^E, and for material temperature at cell i, F_i^T, are,

$$F_i^E=\int_{vol,i}\left[\frac{E_i^{n+1}-E_i^n}{\Delta t}-\frac{\partial}{\partial x}\left(D_{\mathrm{r}}^{n+1}\frac{\partial E^{n+1}}{\partial x}\right)_i-\sigma_{\mathrm{a}}^{n+1}((T_i^{n+1})^4-E_i^{n+1})\right]\mathrm{d}vol_i, \tag{20}$$

$$F_i^T=\int_{vol,i}\left[\frac{T_i^{n+1}-T_i^n}{\Delta t}+\sigma_{\mathrm{a}}^{n+1}((T_i^{n+1})^4-E_i^{n+1})\right]\mathrm{d}vol_i. \tag{21}$$

The goal of a nonlinear iterative method is to drive F_i^E and F_i^T towards zero at each finite volume. It should be noted that in the limit of nonlinear convergence, both fully implicit methods, FIP and NK, must give the same answer since they are solving the same nonlinear problem.

4. Jacobian-free Newton–Krylov method

In this section we provide a detailed description of the proposed nonlinear iterative method. The discussion of Newton's method is standard. We will give some description of a Krylov-based linear iterative method in order to elucidate the Jacobian-free aspect of the proposed method. It is not our intention to give a detailed description of Krylov methods, for which we would recommend the following texts [17, 18].

Newton's method requires the solution of the linear system

$$\mathbf{J}^k \delta \mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k), \qquad \mathbf{u}^{k+1} = \mathbf{u}^k + \delta \mathbf{u}^k, \tag{22}$$

where $\mathbf{J}$ is the Jacobian matrix, $\mathbf{F}(\mathbf{u})$ is the nonlinear system of equations, $\mathbf{u}$ is the state vector, and k is the nonlinear iteration index. For our problem specifically we have,

$$\mathbf{F}(\mathbf{u}) = \{F_1^E, F_1^T, F_2^E, F_2^T, \ \ldots\ , \mathrm{F}_i^E, F_i^T, \ \ldots\ , F_N^E, F_N^T\}, \tag{23}$$

and,

$$\mathbf{u} = \{E_1, T_1, E_2, T_2, \ \ldots\ , E_i, T_i, \ \ldots\ , E_N, T_N\}, \tag{24}$$

where i is the finite volume index, and N is the number of finite volumes. In vector notation, the (i, j)th element of the Jacobian matrix is,

$$J_{i,j} = \frac{\partial F_i(\mathbf{u})}{\partial u_j}. \tag{25}$$

Forming each element of $\mathbf{J}$ requires taking analytic or numerical derivatives of $F_i(\mathbf{u})$ with respect to u at each grid point. This can be both difficult and time consuming. Especially for problems with flux-limiting, and problems for which cross-sections and specific heats are in tabular form.

4.1. Matrix-free approximation

The Generalized Minimal RESidual (GMRES) algorithm is used to solve Eq. (22). GMRES (or any other Krylov method such as conjugate gradients) defines an initial linear residual, $\mathbf{r}_0$ given an initial guess, $\delta \mathbf{u}_0$,

$$\mathbf{r}_0 = -\mathbf{F}(\mathbf{u}) - \mathbf{J}\delta \mathbf{u}_0. \tag{26}$$

Note that the nonlinear iteration index, k, has been dropped. This is because the GMRES iteration is performed at a fixed k. We typically take $\delta \mathbf{u}_0$ equal to zero. The lth GMRES iteration minimizes $\|\mathbf{J}\delta \mathbf{u}_l + \mathbf{F}(\mathbf{u})\|_2$ with a least squares approach. $\delta \mathbf{u}_l$ is constructed from a linear combination of the Krylov vectors (search directions) $\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, (\mathbf{J})^2\mathbf{r}_0, \ \ldots\ , (\mathbf{J})^{l-1}\mathbf{r}_0\}$, which were constructed during the previous $l-1$ GMRES iterations. This linear combination of Krylov vectors can be written as

$$\delta \mathbf{u}_l = \delta \mathbf{u}_0 + \sum_{j=0}^{l-1} \alpha_j (\mathbf{J})^j \mathbf{r}_0, \tag{27}$$

where evaluating the scalars α_j is part of the GMRES iteration. Upon examining Eq. (27) we see that GMRES requires the action of the Jacobian only in the form of matrix–vector products, which

can be approximated by [2];

$$\mathbf{J}\mathbf{v} \approx [\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})]/\varepsilon, \tag{28}$$

where $\mathbf{v}$ is a Krylov vector (i.e. one of $\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, (\mathbf{J})^2\mathbf{r}_0, \ldots, (\mathbf{J})^{l-1}\mathbf{r}_0\}$), and ε is a small perturbation.

Eq. (28) is a first-order Taylor series expansion approximation to the Jacobian, $\mathbf{J}$, times a vector, $\mathbf{v}$. For illustration consider the two coupled nonlinear equations $F_1(u_1, u_2) = 0$, $F_2(u_1, u_2) = 0$. The Jacobian for this problem is

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial F_1}{\partial u_1} & \dfrac{\partial F_1}{\partial u_2} \\ \dfrac{\partial F_2}{\partial u_1} & \dfrac{\partial F_2}{\partial u_2} \end{bmatrix}.$$

Working backwards from Eq. (28), we have

$$\frac{\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\varepsilon} = \begin{pmatrix} \dfrac{F_1(u_1 + \varepsilon v_1, u_2 + \varepsilon v_2) - F_1(u_1, u_2)}{\varepsilon} \\ \dfrac{F_2(u_1 + \varepsilon v_1, u_2 + \varepsilon v_2) - F_2(u_1, u_2)}{\varepsilon} \end{pmatrix}.$$

Approximating $\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{v})$ with a first-order Taylor series expansion about $\mathbf{u}$, we have;

$$\frac{\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\varepsilon} \approx \begin{pmatrix} \dfrac{F_1(u_1, u_2) + \varepsilon v_1 \dfrac{\partial F_1}{\partial u_1} + \varepsilon v_2 \dfrac{\partial F_1}{\partial u_2} - F_1(u_1, u_2)}{\varepsilon} \\ \dfrac{F_2(u_1, u_2) + \varepsilon v_1 \dfrac{\partial F_2}{\partial u_1} + \varepsilon v_2 \dfrac{\partial F_2}{\partial u_2} - F_2(u_1, u_2)}{\varepsilon} \end{pmatrix}.$$

This expression can be simplified to

$$\begin{pmatrix} v_1 \dfrac{\partial F_1}{\partial u_1} + v_2 \dfrac{\partial F_1}{\partial u_2} \\ v_1 \dfrac{\partial F_2}{\partial u_1} + v_2 \dfrac{\partial F_2}{\partial u_2} \end{pmatrix} = \mathbf{J}\mathbf{v}.$$

This matrix-free approach, besides its obvious memory advantage, has many unique capabilities. Namely, Newton-like nonlinear convergence without *forming* or *inverting* the true Jacobian.

To complete the description of this technique we provide a prescription for evaluating the scalar perturbation. In this study ε is given by

$$\varepsilon = \frac{1}{N \|\mathbf{v}\|_2} \sum_{m=1}^{N} b|u_m|, \tag{29}$$

where N is the linear system dimension and b is a constant whose magnitude is approximately the square root of machine roundoff ($b = 10^{-5}$ for most of this study).

4.2. Preconditioning

Traditionally one uses a simple iterative method as a preconditioner to GMRES. The purpose of preconditioning is to efficiently cluster eigenvalues of the iteration matrix, which in turn will reduce the required number of GMRES iterations. The fact that a matrix is formed for the purpose of preconditioning is the motive for calling this method Jacobian-free, and not matrix-free. As we will show, the preconditioning matrix which is formed is much simpler than the true Jacobian of the system.

We employ right preconditioning and thus we are solving,

$$(\mathbf{J}\mathbf{M}^{-1})(\mathbf{M}\delta\mathbf{u}) = -\mathbf{F}(\mathbf{u}). \tag{30}$$

$\mathbf{M}$ symbolically represents the preconditioning matrix and $\mathbf{M}^{-1}$ the inverse of preconditioning matrix. In practice, this inverse is only approximately realized through some standard iterative method, and thus we may think of it more as $\tilde{\mathbf{M}}^{-1}$. The right preconditioned matrix-free algorithm is

$$\mathbf{J}\tilde{\mathbf{M}}^{-1}\mathbf{v} \approx [\mathbf{F}(\mathbf{u} + \varepsilon\tilde{\mathbf{M}}^{-1}\mathbf{v}) - \mathbf{F}(\mathbf{u})]/\varepsilon. \tag{31}$$

This process is actually done in two steps;

(1) Solve (iteratively, and not to convergence) $\mathbf{M}\mathbf{y} = \mathbf{v}$ for $\mathbf{y}$
(2) Perform $\mathbf{J}\mathbf{y} \approx [\mathbf{F}(\mathbf{u} + \varepsilon\mathbf{y}) - \mathbf{F}(\mathbf{u})]/\varepsilon$,

Thus only the matrix $\mathbf{M}$ is formed and only the matrix $\mathbf{M}$ is iteratively inverted. There are two choices to be made;

(1) What linearization should be used to form $\mathbf{M}$?
(2) What linear iterative method should be used to solve $\mathbf{M}\mathbf{y} = \mathbf{v}$?

We have already defined $\mathbf{J}$ is the matrix coming from the Newton linearization of our nonlinear system. Our stated goal is to avoid forming this matrix, but to maintain Newton-like nonlinear convergence. We will refer to the matrix coming from the successive substitution, or Picard-type, linearization as $\mathbf{A}$, and $\mathbf{A}$ will be used for $\mathbf{M}$. Thus, the semi-implicit method will be the preconditioner. At this point the difference between performing a Picard or a Newton nonlinear iteration resides solely in our matrix-vector multiply in GMRES. If we wish to perform Newton nonlinear iterations Eq. (31) is used to perform the matrix–vector multiply $\mathbf{J}\tilde{\mathbf{A}}^{-1}\mathbf{v}$. However, to perform a Picard iteration we use a standard matrix–vector multiply for $\mathbf{A}\tilde{\mathbf{A}}^{-1}\mathbf{v}$, and in place of Eq. (22) we are solving,

$$\mathbf{A}^k\delta\mathbf{u}^k = -\mathbf{F}(\mathbf{u}^k), \qquad \mathbf{u}^{k+1} = \mathbf{u}^k + \delta\mathbf{u}^k. \tag{32}$$

Symmetric Gauss–Seidel is used for iteratively solving $\mathbf{M}\mathbf{y} = \mathbf{v}$ (i.e. $\mathbf{A}\mathbf{y} = \mathbf{v}$) for all methods in this study.

As stated, an inexact convergence tolerance is used on linear solve for the FIP and NK methods. This means that the tolerance to which we solve the linear problem, on each nonlinear iteration, is

related to the current nonlinear residual, $\mathbf{F}(\mathbf{u}^k)$, as

$$\|\mathbf{J}^k\delta\mathbf{u}^k + \mathbf{F}(\mathbf{u}^k)\|_2 < \gamma\|\mathbf{F}(\mathbf{u}^k)\|_2. \tag{33}$$

For most of this study we use a constant $\gamma = 1.0 \times 10^{-2}$. Algorithm sensitivity to the choice of both b in Eq. (29), and γ in Eq. (33) will be investigated.

To summarize, this algorithm achieves Newton-like nonlinear convergence solely through a special matrix–vector multiply routine in GMRES. If we use instead a standard matrix–vector multiply, then one nonlinear iteration is the semi-implicit method, and several nonlinear iterations is referred to as successive substitution, or a Picard iteration. The true Jacobian of the system is never formed and never inverted, and thus we refer to the algorithm as Jacobian-free. An alternative way of viewing this algorithm is that an outer iteration of the matrix-free Newton–Krylov method is used to accelerate the nonlinear convergence of the Picard method.

5. Algorithm performance and results

In this section we study the performance of the proposed algorithm on two model problems with the only difference between the problems being the flux-limiter. For problem 1 we have a unit flux of radiation energy impinging on the left side ($L = 0$) of a cold slab of unit width ($L = 1$). 200 uniform finite volumes are use to discretize the problem in space. The initial conditions are $E^0 = 1.0 \times 10^{-5}$ and $T^0 = (E^0)^{0.25}$. We use $\sigma_a = T^{-3}$, and D_r is flux-limited using method one. The problem is run out to time $t = 3.0$. We ramp our time step up through the first 8 time steps in a pre-defined fashion. The first time step is always equal to 0.1 times the final time step and the first 8 time steps are equal to 2 final time steps.

As a first, rough, measure of accuracy we define the front position as the center of the first cell to drop below $T_r = E^{0.25} = 0.1$. Here we define a second-order accurate in time NK method with a time step of 1.0e-4 as a base and measure the deviation of the other runs from this. Details of higher-order accurate time step methods and time step convergence studies will be the subject of an upcoming study. For this base solution the front position was $x = 0.8325$ at time $= 3$. The solutions for T_r and T, at time $= 1$ and 3 are plotted in Fig. 1. Here we can see the sharpness of the front.

Table 1 compares linear solve requirements, nonlinear iterations, and accuracy for the NK, FIP, and SI methods. the L_2 error of the solution is defined as $L_2(\text{Error}) \equiv \|T_r - T_r^{\text{base}}\|_2$. The nonlinear convergence tolerance within a time step is defined as $\|\mathbf{F}(\mathbf{u}^k)\|_2 \leq 1.0 \times 10^{-5}$. The results in Table 1 indicate that there is a reduction in total linear solves to be had by converging the nonlinearities within a time step. This was accomplished by running the simulation at a larger time step, while maintaining strong nonlinear convergence with the NK methods.

Figs. 2 and 3 demonstrate the nonlinear convergence behavior of the NK and FIP methods respectively. The superiority of the NK method is clear. Increasing the time step by a factor of five caused an increase in the average number of nonlinear iterations per time step by only a factor of two for the NK method. However, this same factor of five increase in time step size in the FIP method resulted in a increase in the average number of nonlinear iterations per time step by more than a factor of five. Again, recall that the only difference between these two methods is in the matrix–vector multiply in GMRES.

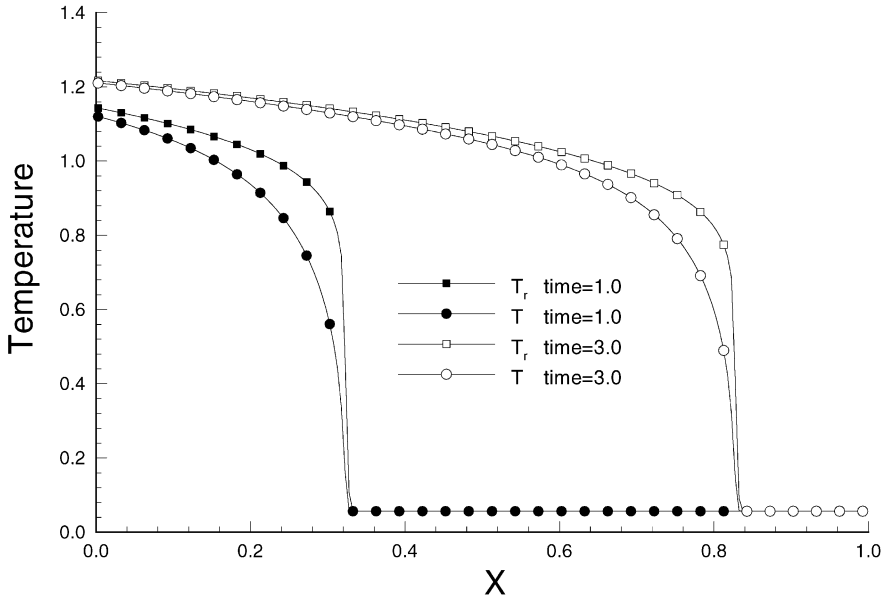

Fig. 1. Base solution for problem 1 at time = 1 and 3.

Table 1
Algorithm performance as a function of time step for problem 1 and time = 3

Method and time step	Number of dt's	Avg. nonlinear Its. per dt	Total linear solves	Front position error (Perct.)	L_2 error
NK, dt = 2e-3	1506	2.1	3163	0.5	0.161
NK, dt = 1e-2	306	3.87	1184	1.0	0.59
FIP, dt = 2e-3	1506	8.6	12,952	0.5	0.162
FIP, dt = 1e-2	306	58	17,748	1.0	0.591
SI, dt = 5e-4	6006	1	6006	− 1.0	0.92
SI, dt = 2e-3	1506	1	1506	− 6.0	2.29
SI, dt = 1e-2	306	1	306	− 20.5	5.11

Fig. 4 compares the results around the front at time = 3, the required number of linear solves are in parentheses. Note that the converged method, NK, is ahead of the "base" answer while the SI solutions are all behind the "base" answer. The FIP results, not plotted, are identical to the NK solutions. The most significant result from this plot is that NK, with a time step of 1.0e-2 and 1184 linear solves, is closer to the base solution (second order accurate in time NK with a time step of 1.0e-4) then the SI method with a time step of 5.0e-4 and 6006 linear solves.

It should be noted that in practice, methods which do not converge nonlinearities within a time step monitor, and control, time step size by looking at maximum relative changes in E. In our results to date we have used a constant time step. For three simulations in Table 1, we report this

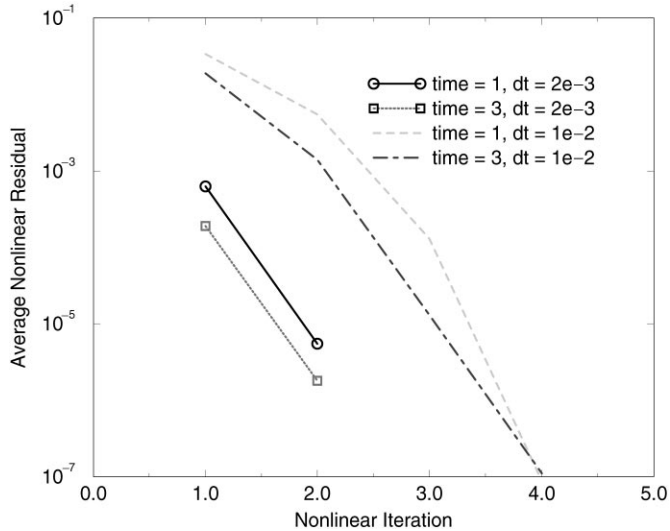

Fig. 2. Convergence plot for NK method on problem 1 at time = 1 and 3, for time steps 1e-2, and 2e-3.

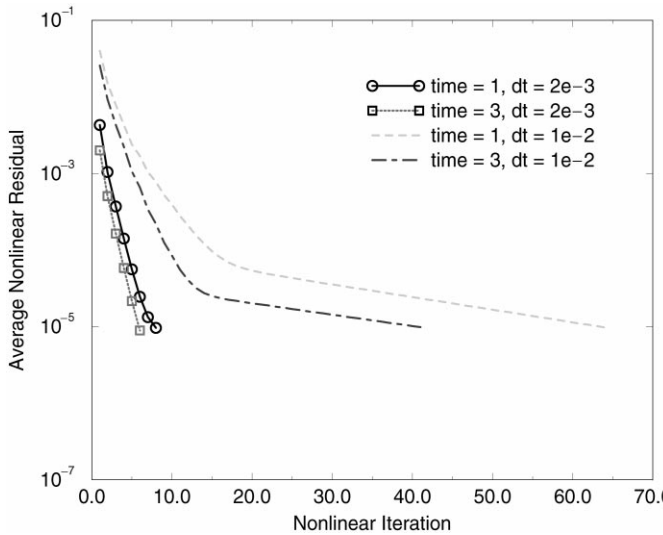

Fig. 3. Convergence plot for FIP method on problem 1 at time = 1 and 3, for time steps 1e-2, and 2e-3.

diagnostic averaged over the last three time steps. Here we use,

$$\frac{\Delta E}{E} = \max_i \left[\frac{|E_i^{n+1} - E_i^n|}{E_i^{n+1} + 1.0 \times 10^{-3}} \right]. \tag{34}$$

For Table 1 we have the following results for $\Delta E/E$ for 3 simulations; 0.807 (NK, dt = 1e-2), 0.232 (SI, dt = 2e-3), and 0.093 (SI, dt = 5e-4). Thus it appears that if the nonlinearities are converged

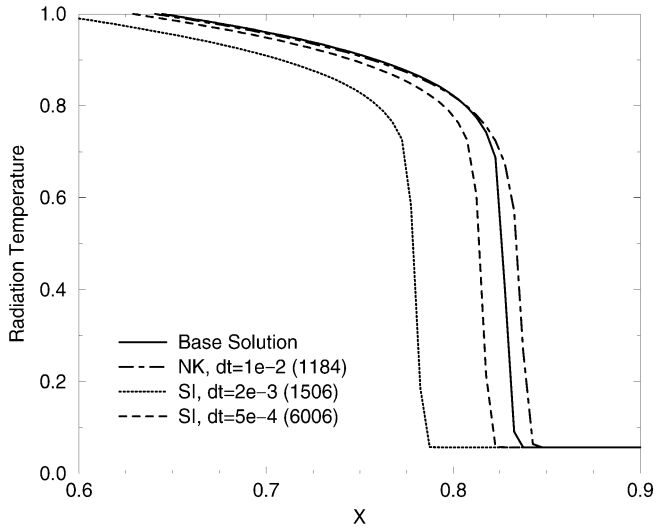

Fig. 4. Comparison of various solutions at time = 3. for problem 1. Linear solve requirements are in parenthesis.

Table 2
Algorithm performance as a function of time step for problem 2 and time = 1

Method and time step	Number of dt's	Avg. nonlinear Its. per dt	Total linear solves	Front position error (Perct.)	L_2 error
NK, dt = 1e-3	1006	2.0	2012	0.0	0.021
NK, dt = 5e-3	206	3.3	680	1.0	0.25
FIP, dt = 1e-3	1006	28.0	28,168	0.0	0.021
FIP, dt = 5e-3	206	82.0	16,895	1.0	0.25
SI, dt = 2.5e-4	4006	1	4006	− 2.0	0.64
SI, dt = 1e-3	1006	1	1006	− 6.5	1.67
SI, dt = 5e-3	206	1	206	− 23.0	3.84

one can sustain a significantly larger change in E per time step, while maintaining increased accuracy.

Problem 2 is identical to problem 1, except that the Levermore flux-limiter is used. This problem is run out to time = 1. Smaller time steps are used here as a result of a faster moving front. Table 2 presents results for this problem and the base solution is plotted in Fig. 5. Again we see that there is a potential significant savings in linear solves by using NK and converging the nonlinearities within a time step. NK is superior to FIP in terms of work required to converge the nonlinearities. NK and FIP provide the same final answer when they both converge the nonlinearities.

Fig. 6 plots nonlinear convergence for the final time step, at two different time step sizes, for both FIP and NK on the same plot. The superior nonlinear convergence of the NK method is again evident. The NK method has produced the same rapidly converging method even with the

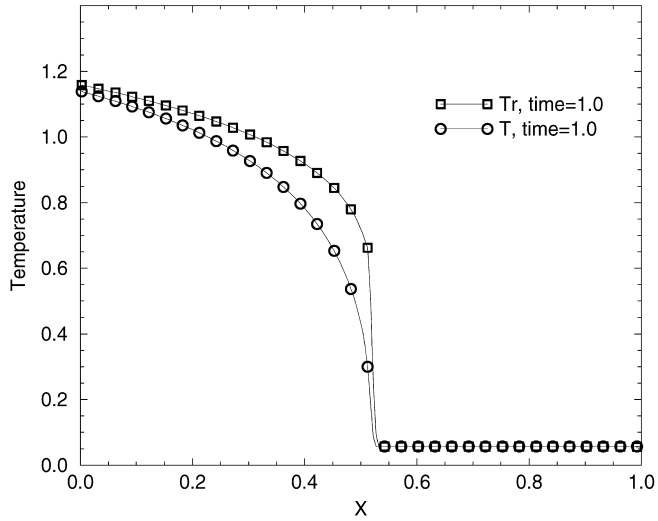

Fig. 5. Base solution for problem 2 at time = 1.

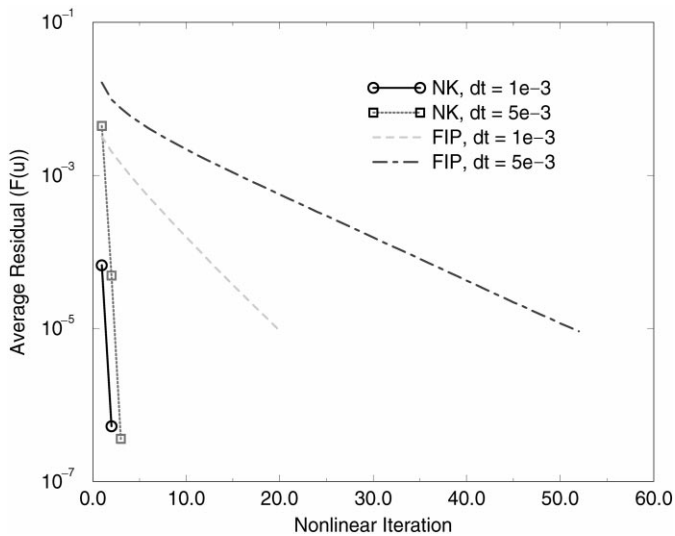

Fig. 6. Convergence plot for FIP and NK methods for problem 2 at time = 1 for time steps 5e-3, and 1e-3.

increased complexity of the Levermore flux-limiter. Recall that the Jacobian elements from the flux-limited diffusion coefficient are never formed.

Finally, Tables 3 and 4 present the sensitivity of the NK method to the choice of ε from Eq. (29) and γ from Eq. (33). These result are for problem 1 run out to time = 1, with a time step of 1e-2. We can see that neither the nonlinear iterations per time step or the linear iterations per time step are sensitive to the value of b, and thus ε. However, while the nonlinear iteration are not very sensitive to the choice of γ, the linear iterations are.

Table 3
Effect of ε (i.e. b) on algorithm performance data for problem 1, dt = 1e-2, up to time = 1, $\gamma = 1 \times 10^{-2}$

Solution method	Avg. Newton its. per time step	Avg. GMRES its. per time step
$b = 1 \times 10^{-4}$	4.21	20.6
$b = 1 \times 10^{-5}$	4.21	20.6
$b = 1 \times 10^{-6}$	4.21	20.6

Table 4
Effect of γ on algorithm performance data for problem 1, dt = 1e-2, up to time = 1, $b = 1 \times 10^{-5}$

Solution method	Avg. Newton its. per time step	Avg. GMRES its. per time step
$\gamma = 5 \times 10^{-2}$	4.47	15.0
$\gamma = 1 \times 10^{-2}$	4.21	20.6
$\gamma = 2 \times 10^{-3}$	4.12	26.4

6. Conclusions

We have presented a new nonlinear solution algorithm for nonequilibrium, flux-limited, radiation diffusion. The method achieves Newton-like, super-linear, convergence without forming or inverting the Jacobian. It appears that there are very real increases in accuracy and/or reductions in total computational effort as a result of converging nonlinearities within a time step using an efficient nonlinear iterative method. It has also been demonstrated that when the nonlinearities are converged within a time step, a simulation can withstand larger values of $\Delta E/E$ while maintaining accuracy.

The importance of the number of GMRES iterations, per Newton iteration, has not been emphasized here because our model problems are 1-D. For this same reason we have used number of linear solves per time step as a measure of effort instead of CPU time. In 2-D we are investigating multigrid based preconditioners in order to limit the number of GMRES iterations per Newton [9]. Here we will also consider CPU time as a more reliable measure of effort.

Future work will investigate the effects of time step control and time step accuracy, along with more complicated model problems.

Acknowledgements

This work was supported under the auspices of the U.S. Department of Energy under DOE contract W-7405-ENG-36 at Los Alamos National Laboratory.

References

[1] Bowers RL, Wilson JR. Numerical modeling in applied physics and astrophysics, Boston: Jones and Bartlett, 1991.
[2] Brown PN, Saad Y. Hybrid Krylov methods for nonlinear systems of equations. SIAM J Sci Stat Comput 1990;11:450–81.
[3] Saad Y, Schultz MH. GMRES: a generalized minimal residual algorithm for solving non-symetric linear systems. SIAM J Sci Stat Comput 1986;7:856.
[4] Knoll DA, McHugh PR. Newton–Krylov methods applied to a system of convection-diffusion-reaction equations. Comput Phys Commun 1995;88:141–60

[5] Knoll DA, McHugh PR, Keyes DE. Newton–Krylov methods for low mach number compressible combustion. AIAA J 1996;34(5).
[6] Knoll DA, McHugh PR. Enhanced nonlinear iterative techniques applied to a nonequilibrium plasma flow. SIAM J. Sci. Comput 1988;19(1).
[7] Mousseau VA, Knoll DA. Fully implicit kinetic solution of collisional plasmas. J Comput Phys 1997;136:308–23.
[8] Kelly CT. Iterative methods for linear and nonlinear equations, Philidelphia: SIAM Frontiers in Applied Mathematics, 1995.
[9] Rider WJ, Knoll DA, Olson GL. A multigrid Newton–Krylov method for multimaterial equilibrium radiation diffusion. J Comput Phys Los Alamos National Laboratory Report, LA-UR-98-2153 (submitted).
[10] D'Amico M. A Newton–Raphson approach for nonlinear diffusion equations in radiation hydrodynamics. JQSRT 1995;54:655.
[11] Stone JM, Mihalas D, Norman ML. Astrophys J (Suppl.) 1992;80:819.
[12] Szilard RH, Pomraning GC. Numerical transport and diffusion methods in radiative transfer. Nucl Sci Eng 1992;112:256.
[13] D'Amico M, Pomraning GC. Treatment of nonlinearities in flux-limited diffusion calculations, JQSRT 1993;49:457.
[14] Winslow AM. Nucl Sci Eng 1968;32:101.
[15] Alme ML, Wilson JR, Astrophys J 1974;194:147.
[16] Levermore CD, Pomraning GC. A flux-limited diffusion theory. Astrophys J 1981;248:321.
[17] Saad Y. Iterative methods for sparse linear systems. PWS Boston: Publishing Company, 1996.